Usage Intelligence

Revulytics

Revulytics Usage Intelligence

Best Practices Overview

Version 1

Table of Contents

Introduction 3
- Overview & Purpose 3
- Development Resources 3
- Additional Resources 3

Product Registration 3

SDK Configuration 4
- Basic Configuration & Ordering 4
- Important Note – Session Duration 4
- Configuration File 5
  - Creating the Config File 5
  - Tracking Multiple User Profiles Per Installation 5
  - Managing Config Files During Software Version Updates 5
- Server-Side Tracking & Multi-Session 6
- Sync 6
- Offline Tracking 6
- Opt-in/out 7

Event Tracking & Custom Properties 7
- Intro: Events vs. Properties 7
- Event Tracking 8
  - Overview 8
  - Deciding Which Events to Track 8
  - Naming Convention 8
  - Event Reports 9
  - Event Tracking API Calls 9
- Custom Properties 10

ReachOut 10

Licensing Key Tracking & Management 10

Integration Checklist 11

About Revulytics 12

Introduction

Overview & Purpose

This is an introductory guide for integrating the Revulytics Usage Intelligence (RUI) SDK into applications. The target audience is software developers charged with performing the SDK integration or product managers overseeing the RUI integration. This guide is not comprehensive API documentation.

Development Resources

Basic implementation of the RUI SDK requires approximately30 minutes of integration with approximately10 lines of code. This basic implementation includes only a list of out-of-the-box metrics. For more comprehensive integration and use of advanced RUI SDK features,

Revulytics recommends reserving 2 – 3 developer days for integration. This target will vary depending on the level of functionality desired (including number of tracked features) and the organization's build, testing, and QA process.

Additional Resources

For additional information, the following resources are available:

Revulytics Devzone – Resources for development (SDK download, API documentation, etc.)

KBase and FAQs - knowledgebase containing answers to commonly asked questions on Usage Intelligence

Blog Posts – Getting Started

Product Registration

Before using the Revulytics Usage Intelligence Software Analytics service or integrating the Revulytics Usage Intelligence SDK, you must first register an account by visiting: https://www.revulytics.com/register. After registering a username and creating a new product account for tracking your application, the Product ID, CallHome URL, and AES Key information is accessible from the Administration page (within the Usage Intelligence dashboard). The latest RUI SDK is also available here.

SDK CONFIGURATION

Basic Configuration & Ordering

When integrating the Revulytics SDK, some API calls are necessary to initialize and configure the Revulytics SDK. Specific ordering of these API calls is required for the Revulytics SDK to successfully initialize. The following is the order of API operations:

1. Initialize SDK object – create an instance of the RUI SDK

2. CreateConfig – Creates a configuration for the RUI SDK instance. A configuration is passed on a file, specified by configFilePath, productID, and appName.

3. SetProductData – Sets or clears the product data. **NOTE:** The product data must be set every time the RUI SDK instance is run.

4. Advanced Integration (Optional)

- **Custom Properties –** Used to collect custom installation metrics. For more information, please see the section Custom Properties.
- **License properties –** Used to collect and/or manage licensing data. For more information, please see the section Licensing.
- **OptOut –** This is an API call that is necessary for any machine opting-out of data collection. For more information, please see the section Opt-in/out.

5. StartSDK – Starts the RUI SDK. StartSDK must be paired with a call to StopSDK.

Additionally, when the application is exiting, the following API call must be made:

- **StopSDK –** Stops the RUI SDK that was started with StartSDK. This function should always be called at the exit point of your application

Important Note – Session Duration

When an application runs for a very short time the SDK will not have enough time to sync with the Revulytics server and create a configuration file.

For automated QA processes, we recommend adding a 10 second “sleep” function between the StartSDK and StopSDK calls. This delay is necessary because the Revulytics SDK needs time to sync with the server. However this delay is not required for normal applications with a runtime duration of more than 10 seconds in everyday use.

CONFIGURATION FILE

Creating the Config File

When your software is first run, the Revulytics Usage Intelligence SDK creates a "ruiconfig*.cfg" file on disk. The location of this file is set through the CreateConfig function. The application must have write permissions to the folder specified. Normallythis path is set to the same location where your application stores data/config/settings.

On some machine configurations (especially machines running Microsoft Windows Vista or higher with UAC enabled), if the user does not have administrator privileges, the application will be denied access to the 'Program Files' directory. Therefore, Revulytics recommends avoiding the default path if your application runs from the Program Files directory.

The config file should only be used by the Revulytics SDK. Your application should not modify or rely on the contents of this file.

Tracking Multiple User Profiles Per Installation

If the application supports multiple user profiles and each user needs to be tracked separately, then a unique folder must be created for every user when calling the CreateConfig method. Using independent folders allows the SDK to generate a unique fingerprint for every user and track them as independent installations. This can be a folder within the user's OS profile or a folder created manually within the application data directory. The folder name could include a unique username/ID/hash identifying that user.

Managing Config Files During Software Version Updates

When a user upgrades product version, we recommend using the same configuration file across multiple product versions. This allows the Revulytics SDK to track an installation across product versions. This can be handled in one of two ways:

- Using the application data path that will not change during the upgrade process (e.g. ProgramData, AppData, etc.)
- Having the application upgrade process handle moving the configuration file.

In the case of an application allowing for parallel installs (i.e. multiple versions installed simultaneously), never share the same configuration file between two installations. Revulytics does not support tracking multiple installations using a single configuration file because each configuration should correspond to its own installation.

The configuration file must not be copied between installations and users, both on the same or different computers. This causes the client fingerprint verification to fail and the Revulytics server will not track usage statistics for that client. Always allow the RUI SDK API to create the config and log files for new installations.

Server-Side Tracking & Multi-Session

Most applications do not require multi-session tracking. Multi-session tracking is used when a servier-side application runtime may host multiple user sessions.

For more information, please see the following Revulytics K-Base article.

For the use case of tracking multiple users for a server-side application, please contact support.

Sync

The Revulytics SDK syncs with the Revulytics server when the calls to StartSDK and StopSDK are made. Sync also happens automatically every 20 minutes while the application is running.

The Reuvlytics SDK also supports manual sync. Manual sync is a complete sync of user data with the server. Manual sync should be used when "ReachOutOnAutoSync" is set to false. For this use case, the sync call is used to retrieve ReachOut messages that can be displayed at specific points in the user's workflow.

Note that manual sync has a 5-minute threshold. If a manual sync is initiated too quickly after another sync has occurred the sync will not be done.

Offline Tracking

Revulytics Usage Intelligence caches tracking data inside a log file named ruilog.log. This file is created after an initial successful sync with the Revulytics server. After this file has been created, when an installation cannot connect to the Revulytics server for a number of days or runtime sessions, all cached data is retained in this log file until the next successful sync.

Note that offline caching requires an initial successful sync. For machines that are permanently offline, no log file will be created and no offline caching will occur.

The Usage Intelligence SDK creates a new log file for each application runtime session. Each runtime log file has a default maximum size of 1MB. The total combined size of all log files is also limited to 1MB by default.

The maximum log file size limitations can be changed by contacting support.

For more information on the list of configuration and log files maintained by the Usage Intelligence SDK please check out this KBase article.

Opt-in/out

Use this mechanism if a user does not want to send tracking information to Revulytics. OptOut must be called after callingCreateConfig and before RUISDK. StartSDK. When OptOut is called, the SDK sends a message informing the server that this client has opted-out. The opt-out flag on the server will be used for reporting opt-out statistics only.

If OptOut is called before a new registration, the server will never have any data about that installation. If OptOut is called for a previously tracked installation the server retains the data that had been collected in the past.

The SDK will send no further information to the server while the user is opted-out. The application must call OptOut before every startup as long as the user wants to stay opted-out. If this function is not called, then the SDK assumes that the user is opting-in again and will start tracking normally.

EVENT TRACKING & CUSTOM PROPERTIES

Intro: Events vs. Properties

For a developer integrating the Revulytics Usage Intelligence SDK, it is beneficial to conceptually understand the difference between an Event and a Property. Some aspects of a product should be tracked as events and other aspects as properties.

Events are actions that occur at specific points in time.

Example – User clicking on a button

Properties are attributes of a product or application.

Example – Product language

In the object-oriented programming paradigm, an Event is analogous to a method and a Property is analogous to a field.

Revulytics Usage Intelligence gives users the abilities to define and track Events as they are triggered. Examples of Events are:

- User interacting with UI elements such as
  - Button clicks
  - Menu launch
- Feature and function usage
- Application state-change

While developers can define their own Custom Properties, the Usage Intelligence SDK automatically collects a list of Properties by default. Examples of Properties collected by default include:

- Screen resolution
- RAM
- Continent/Country/US State
- Lifetime Number of Sessions
- Operating System

EVENT TRACKING

Overview

The terms “Feature” and “Event” are used interchangeably in the scope of Usage Intelligence. In the dashboard interface, Event-related reports are displayed in the section titled “Feature & Event Tracking”.

Deciding Which Events to Track

When integrating the SDK, some users may attempt to track every function and action that exists in their application.

Revulytics strongly recommends researching and identifying your most important software features. While RUI is capable of tagging thousands of unique features, tagging too many complicates the process of extracting information from dashboard reports.

Upon initial integration, we recommend tracking a conservative number of features. This is typically less than 300 unique features. This will help to lessen the development load while still providing valuable insights.

In the dashboard interface users have the capability to enable/disable tracking for particular features. This allows the dashboard user flexibility when selecting which features are tracked for a specific period of time.

Naming Convention

In Revulytics Usage Intelligence, events are defined by a Category and Name. Events can be segmented by category in the dashboard for reporting.

Revulytics recommends implementing an appropriate naming convention for your tracked Events. Examples of naming conventions are:

CATEGORY	NAME
File Operation	Open
File Operation	Save
File Operation	Print
Install Wizard	Step 1
Install Wizard	Step 2

Two common naming schemes use the following conventions:

category=<function>; name=<interface component>;

...or...

category=<interface component>; name=<function>;

If automated or unexposed processes require tracking. we recommend categorizing these processes together. An example is a button-click UI element which triggers 3 additional automated processes. The button-click operation could be tracked on its own. If the 3 automated processes require tracking, these processes should be categorized together.

Event Reports

In the Usage Intelligence dashboard, many reports are generated using event counts. The following reports can be configured to display event count data for any tracked event. No matter what API call is used for event tracking, the data will be displayed in the following reports:

- **Event Usage Timeline –** This report shows aggregate usage statistics for all tracked events, highlighting usage trends over time.
- **Lifetime Event Usage –** This report shows lifetime usage statistics for all tracked events. Lifetime usage statistics are calculated for each client by counting the amount of times an event was fired during their tracked lifetime (i.e. starting from the first install until the last seen date). This enables you to determine how many times users fired an event throughout their lifetime.

Any tracked events can be used as filter criteria in most RUI dashboard reports and also for targeting in-app messaging campaigns via ReachOut.

Event Tracking API Calls

TrackEvent is the standard event tracking API call and is used to record the fact that something occurred (i.e. an action). This is for any event occurring without additional custom data collection.

When an event occurs, you can additionally collect some form of custom data including numeric/text/name-value pairs. For this, use the following API calls (in place of TrackEvent).

TrackEventNumeric logs an event and adds a custom numeric value. This numeric value is used to compute the aggregate and average “Event Custom Value” table in the Event Usage Timeline report.

TrackEventText logs an event and adds a custom string (with a maximum size limit of 4096 characters). TrackEventCustom logs an event and adds a list of name-value pairs (with a maximum size limit of 128 characters per value). The data collected with TrackEventText and TrackEventCustom is available in the Custom Events dashboard report. This report contains a table view of the most recent custom events and allows a dashboard user to download a daily/monthly archive of custom event data.

For additional information on the event tracking API calls, please consult the Revulytics Usage Intelligence API Documentation.

Custom Properties

Usage Intelligence offers the capability to track additional custom properties or attributes that are related to your software or its environment.

For information on supported types of custom properties, please see the following Revulytics KBase article: What type of data can I store in Custom Properties?

For information on adding Custom Properties to your account, please contact Revulytics Support online.

ReachOut

ReachOut In-App messaging is a powerful marketing tool that allows you to interact with your users while they are engaged with your application. By using ReachOut you can take advantage of the analytics data that you collect from your clients to create personalized campaigns and target your audience based on product, licensing, platform, and usage properties.

For more information, please see the following blog post.

Licensing Key Tracking & Management

For software having its own licensing mechanism, this data can be tracked through client-managed license tracking. SetLicenseData is the API call used to collect external license information.

Revulytics Usage Intelligence gives you the ability to manage your license key status via the product dashboard. The application queries the server to determine the status of a particular license key. This is server-managed license tracking.

For more information, please consult the Revulytics Usage Intelligence API Documentation.

INTEGRATION CHECKLIST

☐ Verify that you have configured the correct order of API calls for basic integration:

1. Initialize SDK object
2. CreateConfig
3. SetProductData
4. Custom Properties **(Optional)**
5. License properties **(Optional)**
6. OptOut **(Optional)**
7. StartSDK
8. StopSDK

Application configuration and configuration directory. Verify the following:

- ☐ Config path name: _____
- ☐ Application has “write” permissions to the config file directory
- ☐ Application supports tracking multiple installs on the same physical machine **(Optional)**
  - ☐ Each installation has its own config directory.
- ☐ For version upgrades, application uses the same config file
  - ☐ Application handles moving the configuration file across product versions

  ...or...
  - ☐ Does your application use same config directory across versions?
- ☐ Application allows for parallel installs (multiple versions simultaneously) **(Optional)**
  - ☐ Parallel installs **DO NOT** share the same configuration file

☐ Application is server-side and supports multiple user sessions simultaneously **(Optional)**

- ☐ Multi-session mode is implemented

☐ Application has opt-out mechanism **(Optional)**

- ☐ Application calls OptOut before every startup as long as the user wants to stay opted-out

☐ Approximate number of unique features tracked: _____

- ☐ Determine a standard naming convention which groups event names by product feature or UI location
- ☐ Application logs a single event each time a tracked feature is used. Ensure the same event is not duplicated by different components.

☐ Application uses custom properties **(Optional)**

☐ Application uses ReachOut **(Optional)**

☐ Application uses license tracking **(Optional)**

- ☐ Application uses client-managed license tracking

  ...or...
- ☐ Application uses server-managed license tracking

About Revulytics

Revulytics offers cloud-based software usage analytics that give software producers deep visibility into how their products are being used and misused, providing them with actionable intelligence to generate revenue, optimize product development, and make data-driven decisions across their business. Its compliance analytics solution and turnkey services are used by leading software vendors to increase license revenue and globally reduce software piracy. Its software usage analytics solution provides valuable insight into product usage and environments, enabling product managers and developers to build better products. Revulytics software usage analytics has supported customer compliance programs generating more than $2.1 billion in new license revenue since 2010.

Find out more about Revulytics at:
www.revulytics.com or call +1 (781) 398-3400